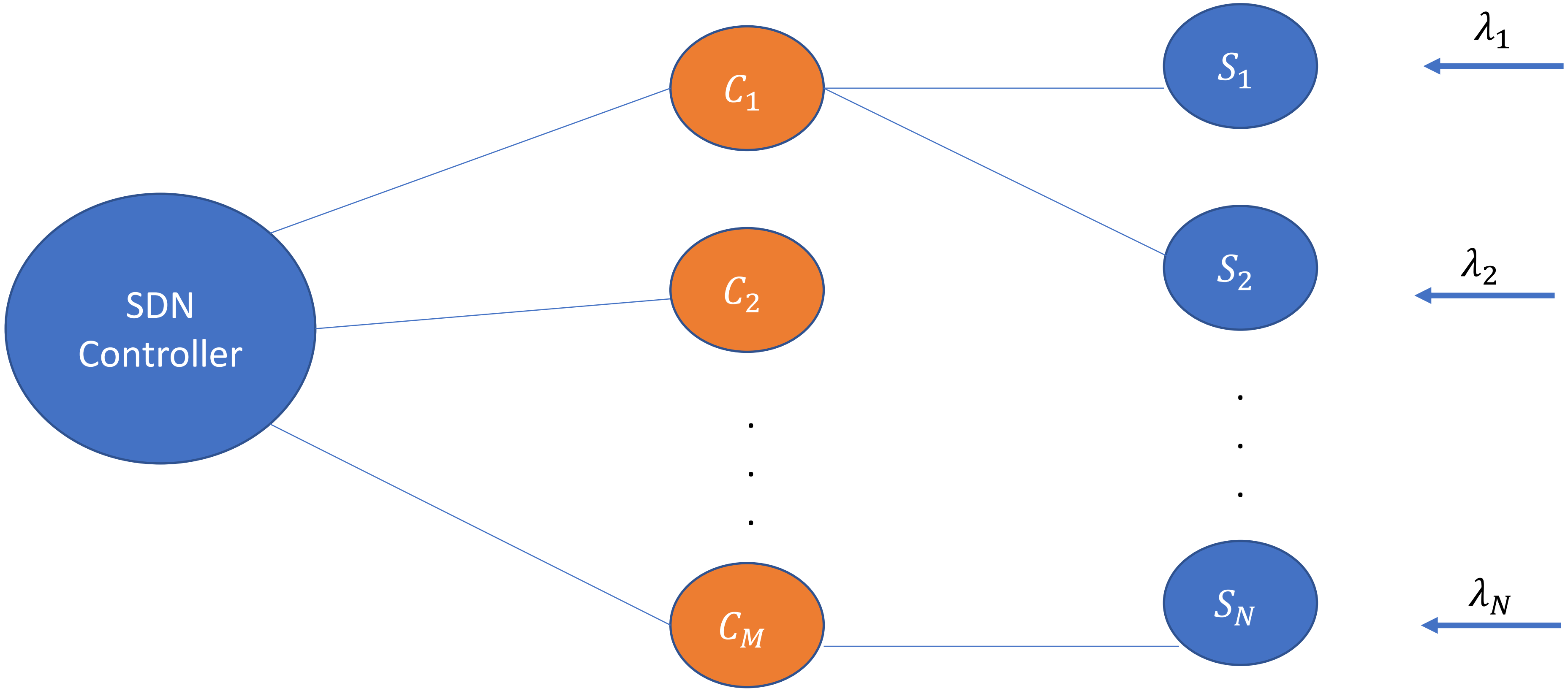
SDN Flow Control



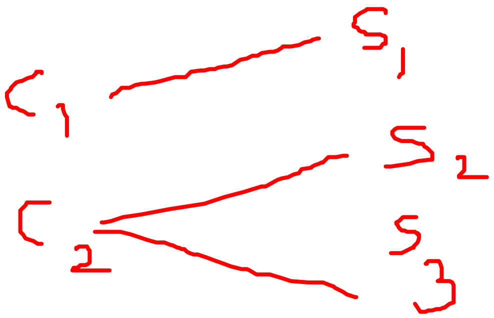
SDN Controller
C_1
C_2
C_M
S_1
S_2
S_N
λ_1
λ_2
λ_N

Model and Assumptions

- Whenever there is an arrival of flow, we need to take a decision
- Flow arrives at switch i as a Poisson process with parameter λ_i
- Service time of each flow is exponentially distributed with parameter μ
- State ={Number of flows in C_1,……, Number of flows in C_N}
- Action space for C_i={Do nothing, Add one switch to C_i, Remove one switch from C_i}

such that all switches are accommodated in the revised allocation after the action is chosen

Model



- Reward= $r(s, a)$ = Some fairness metric after action a is chosen in state s
- Each switch has some limited capability to process flows. Hence, we can assume that the maximum number of flows supported by each switch is N_F.
- Transition probability
  - Example, say 2 controllers and 3 switches, $N_F = 2$
  - Under the action do nothing, $p_{ss'}(a) = \lambda_1/(\sum_{i=1}^{3} \lambda_i + 6\mu)$

    if $s = (i, j)$ and $s' = (i + 1, j)$

$\lambda_2 + \lambda_3$

$(i, j+1)$

Model

- Cost: $c(s,a) = 0$ if action is do nothing

  $c(s,a) = number\ of\ switch\ exchanges\ between\ controllers$ else

  Objective : Maximize the discounted total reward over infinite horizon

  Subject to total discounted cost over infinite horizon

  Formulated as constrained Markov Decision Process (CMDP)

  S_{max} = Constraint on average number of switch exchanges between controller

$$\max_{\pi \in U} \lim_{T \to \infty} \sum_{t=0}^{T-1} \alpha^t E_\pi[r(s_t, a_t)]$$

$$subject\ to \lim_{T \to \infty} \sum_{t=0}^{T-1} \alpha^t E_\pi[c(s_t, a_t)] \leq S_{max}$$

Q-learning

- $Q_{n+1}(s,a) = \big(1 - a(n)\big)Q_n(s,a) + a(n)\big(r(s,a) - l_n c(s,a) + \alpha \max_{a'} Q_n(s',a')\big)$
- $l_{n+1} = \Lambda[l_n + b(n)(S_n - S_{max})]$
- These two schemes need to be triggered whenever there is an arrival of new flow

$S_n = 1 + \alpha \cdot 1 + 0 + \alpha^2 1 + \_\_\_\_\_\_$

- Take $a(n) = \frac{1}{n^{0.6}}, b(n) = \frac{1}{n}$
- Λ is a projection operator to keep the Lagrange multiplier l_n between $[0, L]$ for a large L